

Students Name

ID:

Section:

Software Development Life Cycle (4+6)

Briefly explain how the iterative approach is used in software development?

In your opinion in which development phase the following documents are produced, briefly explain how:

- i. Requirements documents.
- ii. Use case and sequence models.

Iterative approach has a number of iterations, in each iteration the stages of the development are repeated in order to evolve the system, and reduce the number of flaws as much as possible.

Requirements Documents produced in the system conception phase, because when someone has an idea for a system and wants to sell it to some organization, he need to do some type of brain storming to collect the ~~sys~~ requirements.

Use case and sequence models produced in ~~in~~ the analysis phase in order to understand the system very well and before beginning the design phase.

Students Name \_\_\_\_\_

ID: \_\_\_\_\_

Section: \_\_\_\_\_

Use Case and Sequence Diagrams (15)

(10)

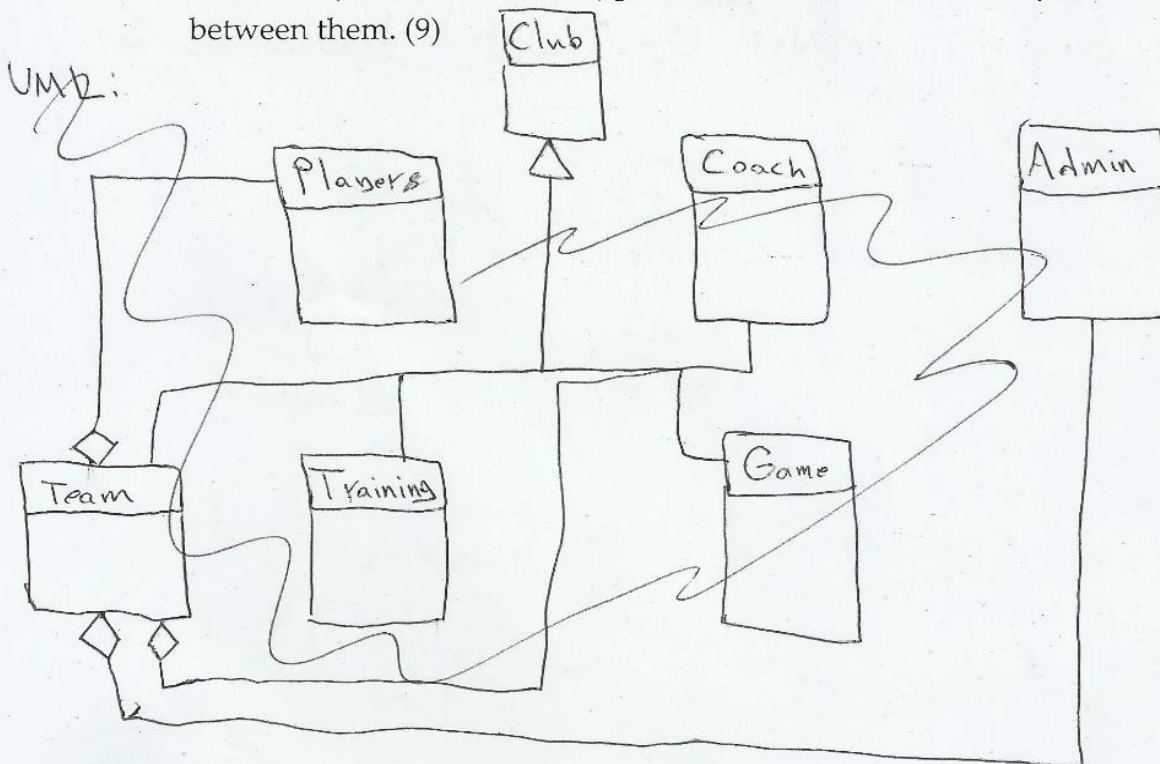
Consider the following requirements for a computerized system:

A sports club would like to record details of its teams' training program and official games. Each team is made-up of players, a coach and an administrator. The club may have many teams for different sports activities. A player, a coach, and an administrator can only be part of one team.

The coach is allowed to set-up his team of players and administrator for a specific sport. He is also responsible to set their training and official games time tables. The players and administrator are allowed to view their training and games time table.

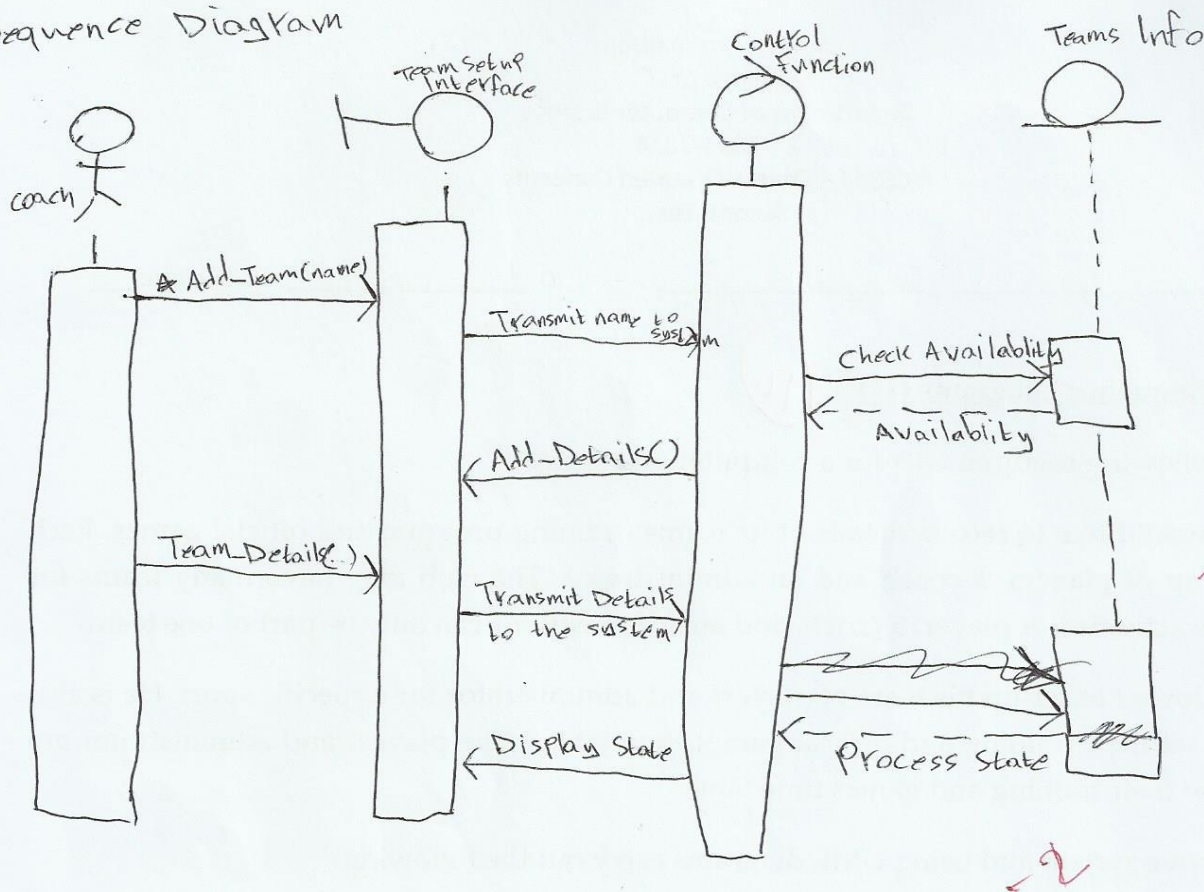
Analyze the above system and using UML diagrams carry out the following:

- Develop all possible use cases for the above system showing all possible relationships between them if any. (6)
- Develop an elaborated sequence diagram for the teams' setup use case showing the various objects involved (of type interface, control, and entity) and the interactions between them. (9)





# Sequence Diagram



Assumption:

Team-Details(~~...~~) includes the required details as a parameters.

(5)

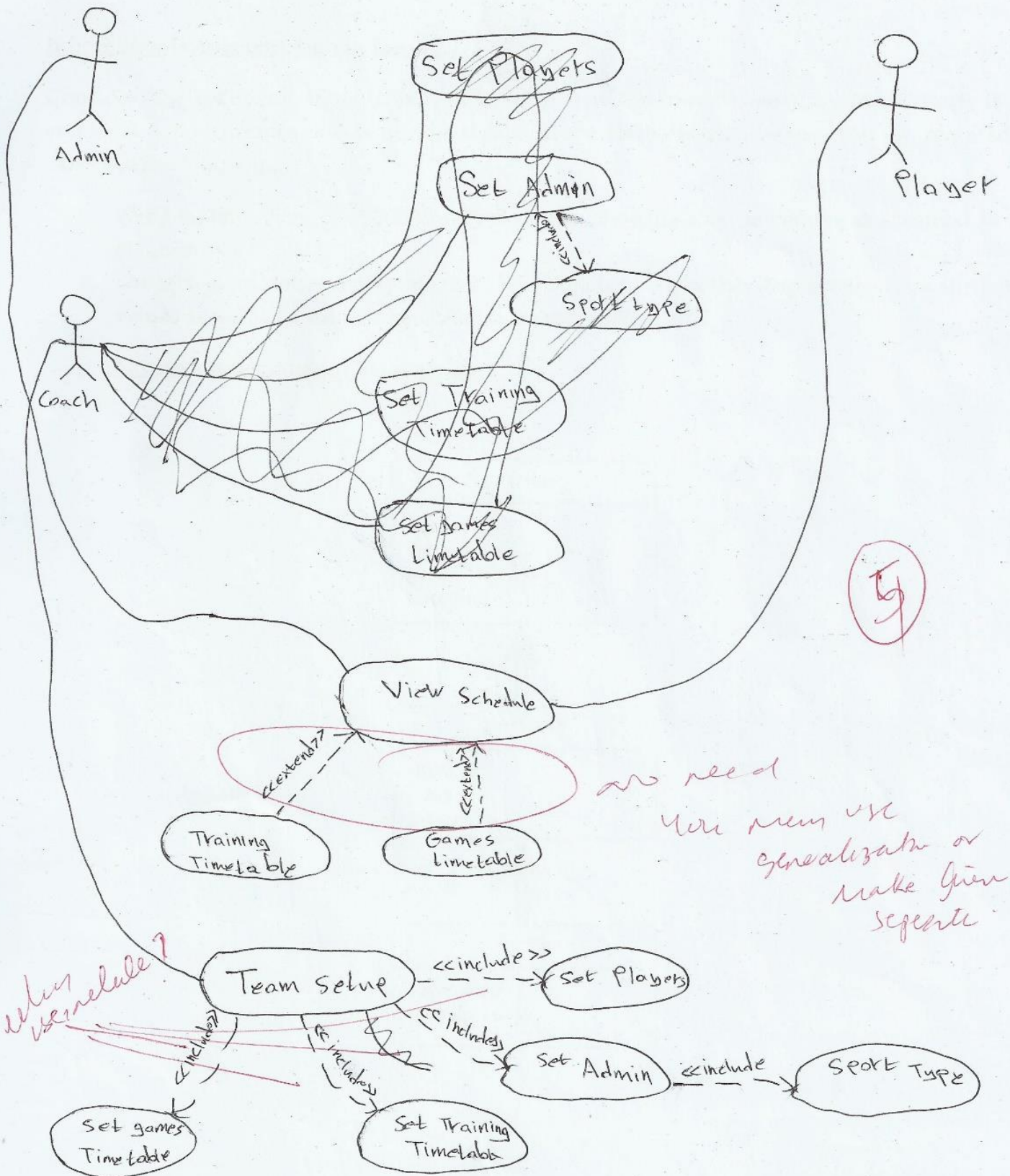
University of Bahrain  
 College of IT  
 Department of Computer Science  
 2<sup>nd</sup> 2013-2014  
 ITCS341 – Object-Oriented Concepts  
 Second Test

Use Case

Students Name \_\_\_\_\_

ID: \_\_\_\_\_

Section: \_\_\_\_\_





Students Name \_\_\_\_\_

ID: \_\_\_\_\_

Section: \_\_\_\_\_

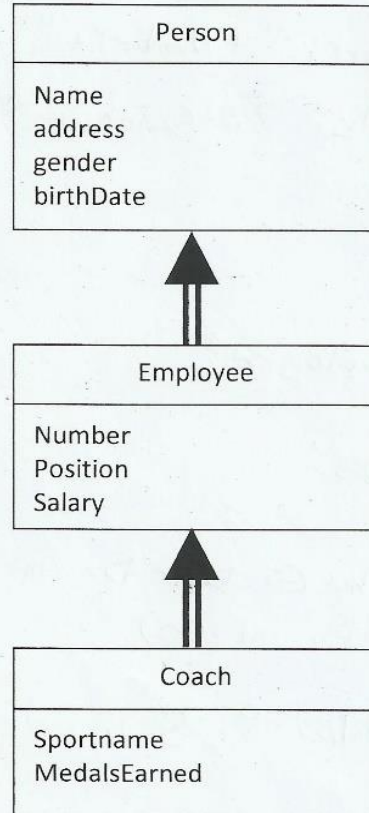
**Inheritance Implementation in Java (15)**

15

Consider the following inheritance relationship between three classes. A team coach is an employee who in return is a person as shown in the class diagram below. You are required to carry out the following:

- Write suitable Java code for the three classes showing their attributes as specified in the diagram. (6)
- For each class, define a polymorphic showDetails function that displays the class attributes' values as well as a suitable constructor function. (9)

No other function need to be specified.



```
class Employee extends Person {
```

```
    private int Number;
```

```
    private String Position;
```

```
    private double Salary;
```

```
    Public
```

```
    Employee(String N, String A, char G, Date D, int No,  
              String P, double S)
```

```
    {  
        super(N, A, G, D)
```

```
        this.Number = No;
```

```
        this.String
```

```
        this.Position = P;
```

```
        this.Salary = S;
```

```
    }
```

```
    public void showDetails()
```

```
    {
```

```
        super.showDetails();
```

```
        System.out.print("Number: " + Number + "\n Position: " + Position +  
                          "Salary: " + Salary + "\n");
```

```
    }
```

```
}
```

```
class Coach extends Person Employee {
```

```
    private String Sportname;
```

```
    private String Medals Earned;
```

```
    private int Count;
```

```
    Coach(String N, String A, char G, Date D, int No, String P, double S,  
          String SP, String ME, int CO)
```

```
    {  
        super(N, A, G, D, No, P, S);
```

```
        this.Sportname = SP;
```

```
        this.M
```

```
        Co = this.Count = CO;
```

```
        this.ME = new
```

```
        this.Medals Earned E = new
```

```
        this.Medals Earned = ME;
```

```
    }
```

University of Bahrain  
College of IT  
Department of Computer Science  
2<sup>nd</sup> 2013-2014  
ITCS341 – Object-Oriented Concepts  
Second Test

Students Name \_\_\_\_\_

ID: \_\_\_\_\_

Section: \_\_\_\_\_

```
class Person {
```

```
    private String Name;
```

```
    private String address;
```

```
    private String char gender;
```

```
    private Date birthDate;
```

```
    public void showDetails()
```

```
    {  
        System.out.print("Name: " + Name + "\nAddress: " + address + "\nGender: "  
            + gender + "\nBirthDate: " + birthDate + "\n");  
    }
```

```
    Person(String N, String Address, char G, Date D)
```

```
    {  
        this.Name = N;  
        this.address = Address;  
        this.gender = G;  
        this.birthDate = D;  
    }
```

```
    }
```

```
}
```

← Cont,



showDetails() {

super.showDetails()

System.out.print("Sport: " + Sportname + "MedalsEarned: " + MedalsEarned);

}

}